CLAIMS

1. A floating point multiplier and accumulator for repetitively performing multiplication and addition operations to create a product of a first operand and a second operand and adding a third operand to the product, comprising:

a multiplier array having a first input and a second input for respectively receiving the first operand and the second operand, and providing a sum and a carry;

multiplexor circuitry coupled to the multiplier array for selectively bit shifting each of the sum and the carry;

shifting circuitry for receiving the third operand and selectively bit shifting the third operand;

adder circuitry coupled to the multiplexor circuitry and the shifting circuitry, the adder circuitry adding the sum, the carry and the third operand to provide at an output thereof a mantissa portion of a resultant operand that has not been normalized or rounded; and

feedback circuitry coupled to the adder circuitry and the multiplier array, the feedback circuitry coupling the mantissa portion of the resultant operand to either the multiplier array or the shifting circuitry to be subsequently used as one of the first operand, the second operand or the third operand, such feedback being done without first performing normalization or rounding, thereby reducing latency associated with creating the product when one of the first operand, the second operand or the third operand has a value that is dependent upon a previous resultant operand calculated by the floating point multiplier and accumulator.

2. The floating point multiplier and accumulator of claim 1 wherein the adder circuitry further comprises:

a carry save adder having a first input for receiving the sum, a second input for receiving the carry, and a third input for receiving the third operand, a first output for providing a sum output, and a second output for providing a carry output; and

a carry propagate adder having a first input coupled to the sum output of the carry save adder, a second input coupled to the carry output of the carry save adder, and an output for providing the resultant operand.

3. The floating point multiplier and accumulator of claim 1 further comprising:

a normalizer coupled to the output of the adder circuitry, the normalizer removing leading edge zeroes from the resultant operand after the resultant operand has been fed back to be used as one of the first operand, the second operand or the third operand in a subsequent calculation of the floating point multiplier and accumulator.

4. The floating point multiplier and accumulator of claim 1 further comprising:

a selective inverter coupled to the output of the adder circuitry, the selective inverter changing a logic value of each bit of the resultant operand in response to determining that the resultant operand has a negative sign, selective inversion occurring prior to feeding the mantissa portion of the resultant operand back to the multiplier array but without normalization or rounding of the resultant operand.

5. The floating point multiplier and accumulator of claim 1 further comprising:

a first register for receiving and selectively storing either the first operand or the resultant operand, the first register being coupled to the first input of the multiplier array, the first register selecting either the first operand or the resultant operand in response to execution of an operational code; and

a second register for receiving and selectively storing either the second operand or the resultant operand, the second register being coupled to the second input of the multiplier array, the second register selecting either the second operand or the resultant operand in response to execution of the operational code.

6. The floating point multiplier and accumulator of claim 1 further comprising:

first control circuitry coupled to the multiplexor circuitry for controlling bit shifting of the sum and carry generated by the multiplier array, the first control circuitry performing predetermined calculations to determine a bit shift amount to shift the sum and carry; and

second control circuitry coupled to the shifting circuitry for controlling an amount of shifting performed by the shifting circuitry.

7. The floating point multiplier and accumulator of claim 1 further comprising exponent generation circuitry for generating an exponent value of the resultant operand based upon an internal exponent value and a number of leading zeroes contained in the resultant operand.

8. The floating point multiplier and accumulator of claim 1 further comprising:

a register for receiving and selectively storing the third operand or the resultant operand; and

a selective inverter coupled to the register for selectively inverting a logic state of each bit position of each value stored in the register based upon a sign of each value, the selective inverter having an output coupled to the shifting circuitry.

9. The floating point multiplier and accumulator of claim 1 wherein the shifting circuitry further comprises a right shifter that selectively bit shifts the third operand or the resultant operand in response to predetermined calculations that determine a bit shift amount, if any, to implement.

10. The floating point multiplier of claim 1 wherein the multiplexor circuitry bit shifts each of the sum and carry by a number of bits that is determined, in part, by a number of leading zero bits of the resultant operand from a previous calculation performed by the multiplier array.

11. In an integrated circuit, a method of repetitively performing multiplication and addition operations with floating point data represented by a first operand, a second operand and a third operand, comprising:

storing the first operand, the second operand and the third operand in storage registers;

coupling the first operand and the second operand to an array multiplier;

multiplying the first operand and the second operand to create a sum and a carry;

selectively bit shifting the sum and the carry based upon: (1) a number of leading zero bits of a resultant operand of a previous calculation; (2) exponent values of the first operand and the second operand; and (3) an exponent value of the resultant operand of the previous calculation;

adding the sum and carry to the third operand to form a cumulative sum;

selectively using the cumulative sum as the third operand for a subsequent multiply/accumulate operation, the cumulative sum not being normalized or rounded prior to such use as the third operand.

12. The method of claim 11 further comprising:

inverting the cumulative sum if a sign of the cumulative sum is negative; and

selectively using the cumulative sum in a subsequent multiply/accumulate operation, in inverted or non-inverted form, as one of the first operand or the second operand when the cumulative sum is not subsequently used as the third operand, the cumulative sum not being normalized or rounded prior to such use as one of the first operand or the second operand.

13. The method of claim 11 further comprising:

generating an exponent value of the resultant operand based upon an internal exponent value generated using exponents of the first and second operands and based upon a number of leading zeroes contained in the resultant operand.

14. In a floating point hardware multiplier and accumulator, a method for repetitively performing multiplication and addition operations to create a product of a first operand and a second operand and adding a third operand to the product, comprising:

coupling a first operand and a second operand to multiplier array circuitry and providing a sum and a carry in response thereto;

selectively bit shifting each of the sum and the carry and selectively adding a plurality of logic zero bits into a portion of the sum and carry in response to control signals created using exponential values of the first operand, the second operand and the third operand and a previous resultant exponential value;

receiving the third operand and selectively bit shifting the third operand;

adding the sum, the carry and the third operand to provide a mantissa portion of a resultant operand that has not been normalized or rounded; and

coupling the mantissa portion of the resultant operand to be subsequently used as one of the first operand, the second operand or the third operand without first performing normalization or rounding, thereby reducing latency associated with creating the product when one of the first operand, the second operand or the third operand has a value that is dependent upon a previous resultant operand calculated by the floating point multiplier and accumulator.

15. The method of claim 14 further comprising:

determining whether the resultant operand is positive or negative and, if the resultant operand is negative, inverting logic state of all bits of the resultant operand prior to using the resultant operand as one of the first operand, the second operand or the third operand in a subsequent calculation.